

1995720964

# Stable Adaptive Neurocontrollers for Spacecraft and Space Robots

Robert M. Sanner\*  
University of Maryland at College Park  
rmsanner@eng.umd.edu

510-63  
47260  
P-12

## Abstract

*This paper reviews recently developed techniques of adaptive nonlinear control using neural networks, and demonstrates their application to two important practical problems in orbital operations. An adaptive neurocontroller is first developed for spacecraft attitude control applications, and then the same design, slightly modified, is shown to be effective in the control of free-floating orbital manipulators. The algorithms discussed have guaranteed stability and convergence properties, and thus constitute viable alternatives to existing control methodologies. Simulation results are presented demonstrating the performance of each algorithm with representative dynamic models.*

## 1 Introduction

Neural networks offer the potential for significantly extending the ability to control complex, poorly modeled dynamic systems. Unfortunately, however, connectionist control efforts often overlook the vast array of tools which have been developed in nonlinear systems theory, including adaptive techniques which are often much less complex than proposed neurocontrol solutions. Moreover, the crucial question of closed-loop stability is often ignored, or treated in an ad hoc fashion in connectionist control applications. Experienced control practitioners are thus often justifiably skeptical about the utility of proposed adaptive neurocontrollers.

\*Dept. of Aerospace Eng. and Space Systems Lab., College Park, MD 20742

Fortunately, however, it is possible to incorporate neural networks into the existing framework of nonlinear control and stability theory, and thereby develop designs which both advance the state of the art and possess guarantees of closed-loop stability and convergence. By uniting the new multivariable adaptive neurocontroller designs of (Sanner&Slotine,1992; Sanner&Slotine,1995) with recent work on adaptive and robust spacecraft attitude controllers (Bach&Paielli,1993; Dwyer&Sira-Ramirez,1988; Egeland&Godhavn,1994; Slotine&Di Benedetto, 1990) we review below the construction of a stable neurocontroller for spacecraft attitude maneuvers. Noting then that the dynamics of a free-floating orbital robot have a structure mathematically similar to rigid spacecraft rotations (Papadopoulos,1990,1991), a similar adaptive neurocontrol methodology can be specified for these space robotic systems. Significantly, free-floating robotic systems cannot be treated in the context of "classic" nonlinear adaptive systems theory, and thus adaptive neurocontrollers represent an important new enabling technology in space robotics.

Section 2 first discusses available nonlinear control techniques for spacecraft attitude maneuvers, then demonstrates how adaptive neural networks can be used to significantly extend these methods when faced with relatively unstructured uncertainty about nature of the torques influencing the motion of the spacecraft. In Section 3, the same neurocontrol design, slightly modified, is shown to be effective in the control of free-floating orbital manipulators. Each section provides a complete specification of the structure of the control and adaptation laws, and provides simulation results which demonstrate the per-

formance of the controller on representative systems.

## 2 Attitude control

### 2.1 Problem Statement

The attitude dynamics of a rigid spacecraft subject to torques applied by gas jet thrusters can be written as (Hughes,1986)

$$\mathbf{H}\dot{\boldsymbol{\omega}} - \mathbf{S}(\mathbf{H}\boldsymbol{\omega})\boldsymbol{\omega} = \boldsymbol{\tau} \quad (1)$$

$$\dot{\mathbf{C}} = -\mathbf{S}(\boldsymbol{\omega})\mathbf{C} \quad (2)$$

where  $\mathbf{H}$  is the constant, symmetric, positive definite spacecraft inertia matrix,  $\mathbf{C}$  is the rotation matrix which describes the attitude of the vehicle with respect to an inertial frame, and  $\boldsymbol{\omega}$  is the angular velocity of the spacecraft with respect to this frame. The vector  $\boldsymbol{\tau}$  represents the torques applied to the spacecraft by its attached attitude control thrusters. In these equations,  $\mathbf{S}$  provides the matrix representation of the cross product operator, so that  $\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a})\mathbf{b}$ , and hence

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Given measurements of the current vehicle attitude and angular velocity, the goal of the attitude control problem is to design a feedback control law for the torques,  $\boldsymbol{\tau}$ , which will ensure that the actual attitude will asymptotically track a desired attitude, defined by

$$\dot{\mathbf{C}}_d = -\mathbf{S}(\boldsymbol{\omega}_d)\mathbf{C}_d, \quad (3)$$

where  $\boldsymbol{\omega}_d$  is a specified desired angular velocity, assumed to be bounded, with a bounded derivative.

While the elements of the direction cosine matrix can be used directly to develop suitable control laws (Bach&Paielli,1993), more compact and computationally efficient algorithms can be developed by instead utilizing the quaternion representation of vehicle attitude. In this formulation, vehicle attitude is specified by a three element vector,  $\boldsymbol{\epsilon}$ , and a scalar parameter,  $\eta$ , collected together into the four element quaternion,  $\mathbf{e}$ , defined so that

$$\mathbf{e} = \begin{bmatrix} \boldsymbol{\epsilon} \\ \eta \end{bmatrix} = \begin{bmatrix} \sin \frac{\varphi}{2} \mathbf{a} \\ \cos \frac{\varphi}{2} \end{bmatrix}.$$

Here  $\mathbf{a}$  is the unit eigenaxis of the rotation from the inertial to the body frame, i.e.  $\mathbf{a} = \mathbf{C}\mathbf{a}$ , and  $\varphi$  is the magnitude of the rotation about this axis (Hughes, 1986). More explicitly, the elements of the quaternion completely determine the rotation matrix  $\mathbf{C}$  through the relation  $\mathbf{C} = \mathbf{R}(\mathbf{e})$ , where

$$\mathbf{R}(\mathbf{e}) = (\eta^2 - \boldsymbol{\epsilon}^T \boldsymbol{\epsilon})\mathbf{I} + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T - 2\eta\mathbf{S}(\boldsymbol{\epsilon}).$$

Finally, in place of (2), the kinematics of the quaternion representing the vehicle attitude is given by

$$\dot{\mathbf{e}} = \mathbf{J}(\mathbf{e})\boldsymbol{\omega} \quad (4)$$

where

$$\mathbf{J}(\mathbf{e}) = \frac{1}{2} \begin{bmatrix} \eta\mathbf{I} + \mathbf{S}(\boldsymbol{\epsilon}) \\ -\boldsymbol{\epsilon}^T \end{bmatrix}.$$

A similar equation defines the evolution of the desired attitude,  $\dot{\mathbf{e}}_d = \mathbf{J}(\mathbf{e}_d)\boldsymbol{\omega}_d$ .

In order to develop a feedback control strategy for this system, an appropriate measure of attitude error must be synthesized. Using the actual and desired rotation matrices, a natural measure for this purpose can be defined as

$$\tilde{\mathbf{C}} = \mathbf{C}\mathbf{C}_d^T. \quad (5)$$

With this definition,  $\tilde{\mathbf{C}}$  is the matrix which transforms a vector in the desired frame to one in the body frame, and in particular, when  $\mathbf{C} = \mathbf{C}_d$ ,  $\tilde{\mathbf{C}} = \mathbf{I}$ . The dynamics of this error measure are easily computed from the actual and desired attitude dynamics

$$\begin{aligned} \dot{\tilde{\mathbf{C}}} &= \dot{\mathbf{C}}\mathbf{C}_d^T + \mathbf{C}\dot{\mathbf{C}}_d^T \\ &= -\mathbf{S}(\boldsymbol{\omega})\tilde{\mathbf{C}} + \tilde{\mathbf{C}}\mathbf{S}(\boldsymbol{\omega}_d) \\ &= -\mathbf{S}(\tilde{\boldsymbol{\omega}})\tilde{\mathbf{C}} \end{aligned} \quad (6)$$

where  $\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega} - \tilde{\mathbf{C}}\boldsymbol{\omega}_d$ .

Alternatively, using the quaternion representation one obtains  $\tilde{\mathbf{C}} = \mathbf{R}(\tilde{\mathbf{e}}) = \mathbf{R}(\mathbf{e}\mathbf{e}_d^{-1})$ , where the inverse of a quaternion is defined as

$$\mathbf{e}^{-1} = \begin{bmatrix} -\boldsymbol{\epsilon} \\ \eta \end{bmatrix}$$

and quaternion multiplication is defined so that  $\mathbf{e}_2\mathbf{e}_1 = \mathbf{U}(\mathbf{e}_1)\mathbf{e}_2$  with

$$\mathbf{U}(\mathbf{e}) = \begin{bmatrix} \eta\mathbf{I} + \mathbf{S}(\boldsymbol{\epsilon}) & \boldsymbol{\epsilon} \\ -\boldsymbol{\epsilon}^T & \eta \end{bmatrix}.$$

Using (4) and (6), the quaternion error dynamics can be computed as

$$\dot{\tilde{\mathbf{e}}} = \mathbf{J}(\tilde{\mathbf{e}})\tilde{\boldsymbol{\omega}}. \quad (7)$$

Note that the quaternion measure of attitude error,  $\tilde{\mathbf{e}}$ , admits the same interpretation as  $\tilde{\mathbf{C}}$ . In particular,  $\tilde{\mathbf{e}}$  is the quaternion corresponding to the attitude of the actual frame with respect to the desired vehicle frame, and when the two frames are aligned  $\tilde{\mathbf{e}}^T = [0, 0, 0, \pm 1]$ .

## 2.2 Conventional fixed and adaptive controller designs

Several authors have exploited the structure of the above equations to develop effective nonlinear control strategies which solve the tracking problem posed (Bach&Paielli,1993; Egeland&Godhavn,1994; Fossen,1992; Paielli&Bach,1993; Wen&Kreutz-Delgado,1991; Wie&Barba,1985; Wie et al.,1989). Most recently, (Egeland&Godhavn,1994), building upon the fundamental results of (Slotine&Li,1987; Slotine&Di Benedetto,1990), have proposed a particularly compact algorithm which is especially amenable to adaptive operation. The current section reviews this new algorithm, in preparation for the neurocontrol extensions considered in the following section.

The algorithm of (Egeland&Godhavn,1994) utilizes the composite error metric

$$\mathbf{s} = \tilde{\boldsymbol{\omega}} + \lambda \tilde{\mathbf{e}} = \boldsymbol{\omega} - \boldsymbol{\omega}_r \quad (8)$$

where

$$\boldsymbol{\omega}_r = \tilde{\mathbf{C}}\boldsymbol{\omega}_d - \lambda \tilde{\mathbf{e}} \quad (9)$$

and  $\lambda > 0$  is an arbitrary positive constant. Provided that the system inertia matrix,  $\mathbf{H}$ , is known precisely, the control law

$$\boldsymbol{\tau}(t) = -\mathbf{K}_D(t)\mathbf{s}(t) + \boldsymbol{\tau}^{nl}(t), \quad (10)$$

where  $\mathbf{K}_D(t)$  is a uniformly positive definite matrix and

$$\boldsymbol{\tau}^{nl} = \mathbf{H}\dot{\boldsymbol{\omega}}_r - \mathbf{S}(\mathbf{H}\boldsymbol{\omega})\boldsymbol{\omega}_r,$$

can then be shown to produce asymptotically convergent closed-loop tracking of any desired attitude trajectory, given by  $\mathbf{e}_d$  and  $\boldsymbol{\omega}_d$ .

Under the more realistic assumption that there is some initial uncertainty about the actual distribution of mass in the spacecraft, the above algorithm

can be modified to continuously tune the nonlinear component  $\boldsymbol{\tau}^{nl}$ , thus adaptively compensating for this uncertainty. Implementation of this modification requires first factoring the nonlinear components of the control law:

$$\begin{aligned} \boldsymbol{\tau}^{nl} &= \mathbf{H}\dot{\boldsymbol{\omega}}_r - \mathbf{S}(\mathbf{H}\boldsymbol{\omega})\boldsymbol{\omega}_r \\ &= \mathbf{Y}(\boldsymbol{\omega}, \boldsymbol{\omega}_r, \dot{\boldsymbol{\omega}}_r)\mathbf{a} \end{aligned} \quad (11)$$

where  $\mathbf{a}$  contains the 6 unique elements of the spacecraft inertia matrix. Using this factorization, but perhaps lacking exact knowledge of the mass properties of the spacecraft, the nonlinear components can be implemented using *estimates*,  $\hat{\mathbf{a}}$ , of the true mass properties,  $\mathbf{a}$

$$\boldsymbol{\tau} = -\mathbf{K}_D\mathbf{s} + \mathbf{Y}\hat{\mathbf{a}}. \quad (12)$$

By then continuously tuning these estimates according to the adaptation law

$$\dot{\hat{\mathbf{a}}} = -\boldsymbol{\Gamma}\mathbf{Y}^T\mathbf{s}, \quad (13)$$

where  $\boldsymbol{\Gamma}$  is a constant, symmetric positive definite matrix controlling the rate of adaptation, (Egeland&Godhavn,1994) show that the resulting closed-loop system is stable, and again guarantees asymptotically perfect tracking of any smooth desired attitude trajectory.

Substantial prior knowledge about the rotational dynamics must be utilized in order to separate the nonlinear functions comprising the elements of  $\mathbf{Y}$ , from the mass parameters  $\mathbf{a}$ ; such a parameterization is readily obtained for the idealized rigid body dynamics of a spacecraft. More complete models of the rotational dynamics, however, may also include a variety of environmental torques, arising from gravity gradients, solar pressure, magnetic fields, and atmospheric drag, to name the more significant sources, which may not readily admit such a convenient parameterization of uncertainty. Indeed, in many cases the actual physics underlying the structure of the environmental torques may be too complex or too poorly understood to provide an explicit, closed-form description of their impact on the rotational dynamics. Moreover, by "hardcoding" into  $\mathbf{Y}$  a description of the expected environment, through the choice of specific functions assumed to model these torques, the system becomes excessively "rigid", incapable of responding appropriately to unexpectedly different environments.

In order to address this more general uncertainty model, the next section reviews techniques whereby the established functional approximation abilities of "neural" networks (Cybenko,1989; Girosi&Poggio,1990; Hornik et al.,1989) can be employed to provide the flexibility necessary to compensate for uncertainty on the actual component *functions* in appearing in the dynamic model.

### 2.3 Adaptive control and neural nets

Incorporating the above sources of environmental torques, a more complete model of the rotational dynamics might thus be

$$H\dot{\omega} - S(H\omega)\omega + E(e, \omega) = \tau, \quad (14)$$

where the vector  $E$  now contains any torques applied to the vehicle by its environment. If the structure of these new torques were known explicitly, by augmenting  $\tau^{nl}$  in (??) with the term  $E(e, \omega)$ , the resulting closed-loop system would again provide asymptotically convergent tracking. However, unlike the situation addressed by the algorithms of the previous section, where there was uncertainty only about the mass properties of the spacecraft, in this section the functional form of the torques appearing in the spacecraft dynamic model, both the rigid body and the environmental torques, is assumed to be completely unknown. The required  $\tau^{nl}$  can hence not be implemented, nor can the above adaptive technique be used to learn the required  $\tau^{nl}$ , since by assumption the prerequisite  $Y\mathbf{a}$  parameterization is unknown or impossible to obtain.

Proceeding similarly to (Sanner&Slotine,1995), consider instead the following alternative representation of the nonlinear component of the required control input:

$$\begin{aligned} \tau^{nl} &= H\dot{\omega}_r - S(H\omega)\omega_r + E(e, \omega) \\ &= M(\mathbf{x})\mathbf{v} \end{aligned} \quad (15)$$

or, in component form,

$$\tau_i^{nl} = \sum_{j=1}^7 M_{i,j}(\mathbf{x}) v_j$$

where  $\mathbf{v}^T = [\dot{\omega}_r^T, \omega_r^T, 1]$ , and for notational convenience, the components of the vehicle state have been

collected into a single vector  $\mathbf{x}^T = [e^T, \omega^T]$ . Unlike expansion (11), which decomposes  $\tau^{nl}$  into a matrix of known functions,  $Y$ , multiplying a vector of unknown constants  $\mathbf{a}$ , this expansion decomposes  $\tau^{nl}$  into a  $3 \times 7$  matrix of unknown functions  $M$ , multiplying a vector of 7 known signals  $\mathbf{v}$ .

Without the ability to determine a  $Y\mathbf{a}$  factorization, an adaptive controller capable of producing the required control input must instead learn each of the 21 unknown component *functions*,  $M_{i,j}(\mathbf{x})$ , as opposed to the conventional model which must learn only unknown *constants*,  $\mathbf{a}$ . In the robotic applications considered in (Sanner&Slotine,1995), the controller implements estimates of these functions using adaptive neural networks. Indeed, since the components of  $S(\omega)H$  are continuous functions of their arguments, if the same also is true of the environmental forces,  $E$ , such networks can be used to uniformly approximate to a chosen accuracy each component function of  $M$  on any closed, bounded subset,  $A$ , of the state space (Cybenko,1989; Girosi&Poggio,1990; Hornik et al., 1989).

Thus, if the functions in  $M$  are sufficiently smooth, a neural network approximation of the form

$$\tau_i^N(t) = \sum_{j=1}^7 \mathcal{N}_{i,j}(\mathbf{x}(t), \mathbf{p}) v_j(t) \quad (16)$$

can accurately approximate the required nonlinear control input for appropriate values of the free network parameters  $\mathbf{p}$ . Here each  $\mathcal{N}_{i,j}$  is an output of a single hidden layer neural network of the form

$$\mathcal{N}_{i,j}(\mathbf{x}, \mathbf{p}) = \sum_{k=1}^N c_{i,j,k} g_k(\mathbf{x}, \xi_k),$$

and the neural approximation theorems ensure that, for several different neural computation models,  $g_k$ , there exist values of the free parameters  $N$ ,  $c_{i,j,k}$  and  $\xi_k$ , which will approximate the continuous functions in  $M$  to a chosen level of uniform accuracy on a compact set  $A$ . In this control setting, defining  $\mathbf{d} = \tau^{nl} - \tau^N$ , one thus has that for proper choice of  $N$ ,  $c_{i,j,k}$  and  $\xi_k$

$$|d_i(\mathbf{x}, \mathbf{v})| \leq \sum_{j=1}^7 \delta_{i,j} |v_j|$$

for any point  $\mathbf{x} \in A$ , where each  $\delta_{i,j}$  is the worst case error of the network approximation to  $M_{i,j}$  on the

set  $A$ . Provided that  $\omega_d(t)$  and  $\dot{\omega}_d(t)$  are bounded, as above, over this subset of the state space, the discrepancy between the “neural” approximation and the required nonlinear terms can thus be made arbitrarily small by appropriate design of the network employed (Sanner,1993; Sanner&Slotine,1995).

The network used in (16) has the 7 components of the vehicle attitude state as its inputs, and 21 outputs representing the approximations to each  $M_{i,j}(\mathbf{x})$ . While in principle, each of the independent network parameters,  $N$ ,  $\xi_k$ , and  $c_{i,j,k}$  could be learned, new theoretical results on constructive neural approximation techniques provide a variety of algorithms for effectively preselecting certain of the network parameters based upon estimates of the *smoothness* of the functions being approximated. For example, for certain radial basis function network models (Broomhead&Lowe,1988; Poggio&Girosi,1990), i.e. networks for which  $g_k(\mathbf{x}, \xi_k) = g(\sigma_k \|\mathbf{x} - \xi_k\|)$  for a positive scaling parameter  $\sigma_k$ , the parameters  $\xi_k$  can be chosen to encode a uniform mesh over the set  $A$  whose spacing is determined by bounds on the significant frequency content of the Fourier transform of the functions being approximated (Sanner&Slotine,1992).

This analysis, and similar constructive techniques, leaves only the specific output weights,  $c_{i,j,k}$ , to be learned in order to accurately approximate the particular functions of the assumed smoothness class which appear in the matrix  $M$ . The following section reviews how the techniques of (Sanner&Slotine,1995, Sanner&Slotine,1992) can be applied to the attitude control problem, specifying a neurocontrol law and adaptation mechanism which can stably learn these required output weights, producing asymptotically convergent tracking of a desired attitude. For a more detailed analysis of this algorithm, including more general adaptation methods, refer to (Sanner&Slotine,1995).

## 2.4 Adaptive neurocontroller designs

Despite their potential, practical implementations of neural computation models are at best capable of providing only *locally approximate* representations of the required control input. Use of such a device in place of explicit, prior knowledge about the dynamic structure thus introduces the unmeasur-

able disturbance,  $\mathbf{d}$ , into the closed-loop dynamic model. Since  $\mathbf{d}$  is generally nonvanishing, the adaptive system must be robust to this perturbation, lest it cause the closed-loop system to become unstable (Reed&Ioannou,1989; Sanner&Slotine,1992).

To accommodate the required robust modifications, first define a set  $A_d \subset \mathcal{R}^7$  containing the trajectories the system must follow, a closed and bounded “nominal operating range”  $A \supset A_d$ , and a smooth modulation function,  $m(t)$ , which is unity outside the set  $A$ , vanishes inside  $A_d$  and otherwise satisfies  $0 < m(t) < 1$ . Notice that  $A_d$  can be chosen as the cartesian product of the four dimensional cube  $[-1, 1]^4$  and a three dimensional cube containing  $\omega_d(t)$  for all  $t$ , since by definition the quaternion components only assume values in  $[-1, 1]$ .

The proposed adaptive control law can then be written as

$$\tau(t) = -K_D(t) \mathbf{s}(t) + m(t) \tau^{sl}(t) + (1 - m(t)) \hat{\tau}^N(t) \quad (17)$$

where the robust *sliding controller* component is  $\tau_i^{sl}(t) = -K_i(\mathbf{x}, t) \text{sgn}(s_i(t))$ , whose gains are chosen, similar to the designs in (Slotine&Li,1991; Dwyer&Sira-Ramirez,1988), so that

$$K_i(\mathbf{x}, t) \geq \sum_{j=1}^7 |M_{i,j}(\mathbf{x}) v_j(t)|.$$

These upper bounds, which can be quite loose, are assumed to be available *a priori*.

Assuming a network architecture has been selected on the basis of the assumed smoothness of the functions required in the control law, the adaptive neural component of the controller is given by

$$\hat{\tau}_i^N(t) = \sum_{j=1}^7 \sum_{k=1}^N \hat{c}_{i,j,k}(t) g_k(\mathbf{x}(t), \xi_k) v_j(t). \quad (18)$$

Building from the results in (Sanner,1993; Sanner&Slotine,1995), (Sanner&Vance,1994) show that the control law (17), (18) coupled with the continuous network learning rule

$$\dot{\hat{c}}_{i,j,k}(t) = \mathcal{P}(-\gamma_{i,j,k} s_i(t) v_j(t) g_k(\mathbf{x}(t), \xi_k), \hat{c}_{i,j,k}(t), \bar{c}_{i,j,k}) \quad (19)$$

will produce a stable closed-loop system and asymptotic tracking of any desired attitude with an ultimate accuracy limited only by the network approximation capabilities,  $\delta_{i,j}$ . Here  $\bar{c}_{i,j,k}$  is an upper

bound on the magnitude of each required output weight, and the projection operator  $\mathcal{P}$  is defined so that  $\mathcal{P}(x, y, z) = (1-m)x$  if  $-z < y < z$ , or if  $y \leq -z$  and  $x > 0$ , or if  $y \geq z$  and  $x < 0$ ;  $\mathcal{P}(x, y, z) = 0$  otherwise.

This robust adaptation mechanism effectively restricts the search for the required weights to a subset of the  $21N$  dimensional weight space, preventing the possibly unbounded "wandering" which can be provoked by the disturbance  $\mathbf{d}$  (Slotine&Li,1991; Reed&Ioannou,1989). The robust controller component,  $\tau^{sl}$ , is a supervisory mechanism which, if required, will stabilize the system in its initial learning phases, smoothly returning the state to its nominal operating range, on which the network is capable of well approximating  $M$ .

## 2.5 Attitude control example

This section demonstrates the performance of the proposed algorithm on a simulated attitude control problem. The spacecraft inertia matrix used in the simulation is

$$H = \begin{bmatrix} 60 & 5 & 0 \\ 5 & 78 & 10 \\ 0 & 10 & 38 \end{bmatrix},$$

and the desired attitude trajectory used to evaluate the controller was specified by

$$\begin{aligned} \omega_{d,1} &= \frac{-3(\cos t + 3\sqrt{3}\sin t)}{8\sqrt{2}r(t)} \\ \omega_{d,2} &= \frac{3(5\cos t - \sqrt{3}\sin t)}{8\sqrt{2}r(t)} \\ \omega_{d,3} &= \frac{3(\sqrt{3}\cos t + \sin t)}{8r(t)} \\ \dot{\mathbf{e}}_d &= \mathbf{J}(\mathbf{e}_d)\omega_d \end{aligned}$$

where  $r(t) = 1 + .2\cos t$ . To implement the control law (17), the tracking error metric  $\mathbf{s}$  is computed using (21) with  $\lambda = 10$ , and the gains  $K_D = 100I$  are used for the linear feedback components.

Given the definition of the desired trajectory, the nominal operating range,  $A_d$ , was chosen as  $A_d = [-1, 1]^4 \times [-1.75, 1.75]^3$ . The neural network,  $\mathcal{N}$ , employed in the control law uses radial gaussian nodes, with  $g_k(\mathbf{x}, \xi_k) = \exp(-\sigma_k\|\mathbf{x} - \xi_k\|^2)$  to approximate the functions in  $M$  on the set  $A = [-1.1, 1.1]^4 \times$

$[-2, 2]^3$ . For simplicity in this simulation, the network was designed assuming that any applied environmental forces are a function of  $\omega$  only. Under these conditions,  $M$  is also a function of  $\omega$  only, and the resulting network requires only the three inputs,  $\omega_i$ , and still 21 outputs,  $\mathcal{N}_{i,j}$ . Using the constructive analysis techniques in (Sanner,1993) to initially fix some of the network structure, each node uses the same scaling parameter,  $\sigma_k = 6$ , and the gaussian "centers"  $\xi_k$  lie on a regular lattice of mesh size  $\Delta = 0.5$  covering the set  $[-2.5, 2.5]^3$ . There are thus a total of 1,331 gaussian nodes and 27,951 output weights which the network must learn in order to accurately approximate the elements of  $M$ .

Each output weight was initialized to zero, simulating an initial total lack of knowledge about the dynamics of the system. During the simulation, these weights were continuously updated according to the learning rule (19) together with the adaptation gains  $\gamma_{i,j,k} = 2.5$  for each  $i, j, k$ . The upper bounds  $\bar{c}_{i,j,k} = 200$  were used to implement the projection mechanism. The modulation function,  $m(t)$ , and sliding controller gains were chosen as in (Sanner&Slotine,1992; Sanner&Slotine,1995). In this particular example, however, the supervisory action of the sliding controller was never needed.

Figure 1 shows the performance of the algorithm using this network, when the spacecraft attitude evolves according to the ideal model (1). After a transient period the attitude tracking errors,  $\tilde{\mathbf{e}}_i$ , are reduced to a small neighborhood of zero, and  $\tilde{\eta}$  converges to near 1, indicating that the spacecraft is asymptotically tracking the desired attitude. For comparison, Figure 2 illustrates the tracking which would be obtained without use of the adaptive network, thus implementing a quaternion "PD" type control strategy. The initial performance of the network is virtually identical to the "PD" algorithm, but the network performance rapidly becomes markedly superior.

Figure 3 shows the performance of the algorithm, using the same network and initialization, when the spacecraft attitude instead evolves according to (14), where the environmental torques are given by

square-law drag terms of the form

$$\mathbf{E}(\boldsymbol{\omega}) = \begin{bmatrix} 8|\omega_1| & 0 & 0 \\ 0 & 15|\omega_2| & 0 \\ 0 & 0 & 25|\omega_3| \end{bmatrix} \boldsymbol{\omega}.$$

Note that while this particular environmental torque is not common in an orbital environment, it is constitutes a significant influence in neutral buoyancy simulation of orbital operations. The large perturbations these representative hydrodynamic torques introduce to the ideal rigid body dynamics provide a significant additional dynamic component which must be learned by the neural network. As Figure 3 shows, however, the ultimate tracking performance obtained in the presence of these torques is virtually identical to that obtained with the unperturbed dynamics, indicating that the network is successfully compensating for the new dynamic components. By comparison, if the adaptive contribution of the network is omitted in the control law, the tracking performance is significantly degraded, as demonstrated in Figure 4.

### 3 Free-floating robot control

#### 3.1 Problem statement and neurocontrol solutions

When a robotic arm is mounted to the front of a submersible or orbital vehicle, the motion of the arm will couple to that of its mobile base. If the base is allowed to rotate as the arm moves, that is, if no torques are directly applied to the base allowing it to resist the induced motion, the resulting robotic system is termed a *free-floating* manipulator. Such systems are especially attractive in space operations, where worksite damage could ensue from use of a propulsion system, and where avoiding the use of reaction mass may make the mission potentially more affordable by reducing launch costs and/or extending the useful life of the system.

A careful analysis of the coupled dynamics of a manipulator arm mounted on a free-floating base shows that the spacecraft attitude states may be eliminated from the coupled equations, resulting in a compact set of differential equations describing the motion of arm joints. These equations have the same the same general form as the equations of motion for

fixed-base manipulators (Papadopoulos,1990,1991), i.e.

$$\mathbf{H}^*(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{F}^*(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{E}^*(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau}_m. \quad (20)$$

In this equation,  $\mathbf{q} \in \mathcal{R}^n$  is an  $n$  vector of manipulator joint angles,  $\mathbf{H}^*$  is a symmetric, uniformly positive definite inertia matrix, and  $\mathbf{F}$  is a matrix accounting for the centripetal and Coriolis forces arising from the arm motions. The vector  $\boldsymbol{\tau}_m$  represents the torque applied by motors at each manipulator joint. Finally,  $\mathbf{E}^*$  again represents the effect of any additional environmental forces.

In addition to the similarities to fixed-base manipulator dynamics, (20) is clearly also quite similar to the spacecraft rotation models examined above. Indeed, formally combining the spacecraft kinematic and dynamic equations produces a differential equation structurally identical to (20) (Slotine&Di Benedetto,1990). It is precisely this structural equivalence which has inspired the recent adaptive attitude control algorithms (Egeland&Godhavn,1994; Fossen,1992; Slotine&Di Benedetto,1990), including the one reviewed above, from the fundamental robotic result presented in (Slotine&Li,1987).

This suggests moreover that the adaptive neurocontroller presented above can also be used to cause the joint angles of a free-floating manipulator to asymptotically track any desired sequence of joint angles,  $\mathbf{q}_d$ . By redefining the tracking error metric

$$\mathbf{s} = \dot{\bar{\mathbf{q}}} + \lambda \bar{\mathbf{q}} \quad (21)$$

where now  $\bar{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$ , (Sanner&Vance,1994) show that the preceding adaptive neurocontrol algorithm indeed provides a stable closed-loop system and asymptotic convergence of the tracking errors to a small neighborhood of zero. In such applications, the network inputs are the states of the robotic arm,  $\mathbf{x}^T = [\mathbf{q}^T, \dot{\mathbf{q}}^T]$ , and the auxiliary signals are  $\mathbf{v}^T = [\ddot{\mathbf{q}}_r^T, \dot{\mathbf{q}}_r^T, 1]$ , where  $\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \lambda \bar{\mathbf{q}}$ .

An additional design simplification can be obtained in these robotic applications by noting that the centripetal and Coriolis forces are quadratic in velocity. If also  $\mathbf{E}^*$  is a function of  $\mathbf{q}$  only, or can be decomposed as  $\mathbf{E}^*(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{E}_0^*(\mathbf{q})\mathbf{f}(\dot{\mathbf{q}})$ , where  $\mathbf{f}(\dot{\mathbf{q}})$  represents a known  $\dot{\mathbf{q}}$  dependence, the neural component of the controller can be chosen as

$$\hat{\boldsymbol{\tau}}_i^N(t) = \sum_j \sum_{k=1}^N \hat{c}_{i,j,k}(t) g_k(\mathbf{q}(t), \boldsymbol{\xi}_k) w_j(t). \quad (22)$$

The new auxiliary signals,  $w_j$ , are respectively the components of  $\ddot{\mathbf{q}}_r$ ,  $[\dot{\mathbf{q}}\dot{\mathbf{q}}_r]$ , and  $\mathbf{f}(\dot{\mathbf{q}})$  (or simply 1 for the latter component if  $\mathbf{E}^*$  is a function of  $\mathbf{q}$  only). The notation  $[\dot{\mathbf{q}}\dot{\mathbf{q}}_r]$  is a shorthand for all possible combinations  $\dot{q}_i \dot{q}_{r,j}$  for each  $i, j = 1, \dots, n$ . With this use of the network in the controller, the adaptation mechanism is modified to be

$$\dot{\hat{c}}_{i,j,k}(t) = \mathcal{P}(-\gamma_{i,j,k} s_i(t) w_j(t) g_k(\mathbf{q}(t), \boldsymbol{\xi}_k), \hat{c}_{i,j,k}(t), \bar{c}_{i,j,k}). \quad (23)$$

Despite the gross structural similarity of the dynamics (20) to both fixed-base manipulator dynamics and the spacecraft rotation models (14),(4), there are important differences in the nature of the functions which appear. In particular, the matrices  $\mathbf{H}^*$  and  $\mathbf{F}^*$  in the free-floating manipulator dynamics are significantly more complex than their counterparts in spacecraft or fixed-base robot dynamics. These matrices are so complex, in fact, that in the face of uncertainty about the manipulator mass properties, the parameterization  $\boldsymbol{\tau}^{nl} = \mathbf{Y}\hat{\mathbf{a}}$  is simply *not possible* even for the ideal ( $\mathbf{E}^* = \mathbf{0}$ ) dynamics of free-floating manipulator systems (Papadopoulos,1990; Sanner&Vance,1994). This is in marked contrast to spacecraft and fixed-base robot dynamics, and provides a specific example of a situation in which the adaptive function approximations implemented by neural networks yield a new solution to an otherwise intractable control problem.

### 3.2 Free-floating robot example

Figure 5 shows a 2 link manipulator attached to a spacecraft, with both spacecraft and arm motion restricted to a single plane. The 3 independent degrees of freedom of the system are  $\theta$ , the orientation of the spacecraft with respect to an inertial reference frame,  $q_1$  and  $q_2$  which respectively describe the relative orientation of the first manipulator link to the spacecraft and the second link to the first link.

For simplicity, the simulation assumes an ideal dynamic model with  $\mathbf{E}^* = \mathbf{0}$  in (20). Figure 6 gives the mass, inertia, and relevant dimensions for the system. The centers of mass of the spacecraft and of each link are located centrally, as indicated in Figure 5. To demonstrate the performance of the proposed neurocontroller, the desired trajectory was  $q_{d,1}(t) = 1.2 \cos(0.8t)$  and  $q_{d,2}(t) = 0.5 \cos(2.1t)$ . Given the definition of the trajectories the system

is required to follow, the set  $A_d$  was chosen as  $A_d = [-1.2, 1.2] \times [-0.5, 0.5] \times [-1, 1] \times [1.05, 1.05]$ , and the nominal operating range,  $A$  was chosen as  $A = [-1.4, 1.4] \times [-0.6, 0.6] \times [-1.1, 1.1] \times [1.2, 1.2]$ .

Using the simplified controller with (22) above, the neural network employed in the control law has the 2 inputs  $q_1(t)$  and  $q_2(t)$ , and 12 outputs. The network used for the simulation again employs radial gaussian nodes in the hidden layer, with each gaussian center arranged on a regular lattice of mesh size  $\Delta = 0.2$  covering the set  $[-2, 2] \times [-1.4, 1.4]$ . Each node again used the same scale factor, here taken as  $\sigma_k = 13$ . There are thus a total of 315 gaussian nodes and 3780 output weights which the network must learn in order to accurately approximate the required control input.

Each output weight was again initialized to zero, and continuously updated according to the learning rule (23) together with the adaptation gains  $\gamma_{i,j,k} = 2$  for each  $i, j, k$ . The error metric  $\mathbf{s}$  is computed using (21) with  $\lambda = 10$ , and the gains  $\mathbf{K}_D = 10\mathbf{I}$  are used for the linear feedback control components. Finally, the modulation function and sliding gains were again computed as in (Sanner&Slotine,1995; Sanner&Vance,1994).

Figure 7 displays the performance of the neurocontroller tracking the specified joint space trajectory. After a brief initial transient, the tracking errors in each joint converge to a small neighborhood of zero. Compare this with the performance of the "PD" controller obtained by omitting the contribution of the adaptive network from the control law. Although initially (before any learning has occurred) the performance of the neurocontroller resembles that of the pure "PD" controller, the neurocontroller gradually reduces the tracking error, eventually achieving worst case error a factor of 20 smaller than those obtained with the PD controller.

## 4 Concluding remarks

High performance control of orbital robots and spacecraft is an essential technology to ensure that these systems will be truly useful in future orbital operations. Most importantly, the accuracy and reliability of the algorithms employed must be assured, even in the face of real-world uncertainty on the physical properties of the system. In this paper we

have demonstrated that, far from academic curiosities, adaptive "neural" networks provide unique solutions for important practical problems in the control of spacecraft and space robots, which otherwise are difficult to solve with established adaptive control techniques. The stability and convergence properties of the algorithms described provide the assurances of reliability and effectiveness needed to make such controllers viable alternatives to existing control algorithms.

## References

- [1] Bach, R., and Paielli, R., "Linearization of attitude-control error dynamics", *IEEE Trans. Aut. Control*, 38(10), 1521-1525.
- [2] Broomhead, D. S., and Lowe, D., "Multivariable functional interpolation and adaptive networks", *Complex Systems*, 2, 321-355, 1988.
- [3] Cybenko, G., "Approximations by superposition of a sigmoidal function", *Math. Cont., Sig. and Sys.*, vol. 2, 303-314, 1989.
- [4] Dwyer, T., and Sira-Ramirez, H., "Variable-structure control of spacecraft attitude maneuvers", *J. Guid., Contr., and Dyn.*, 11(3), pp. 262-270, 1988.
- [5] Egeland, O., and Godhavn, J.-M., "Passivity-based adaptive attitude control of a rigid spacecraft", *IEEE Trans. Aut. Control*, 39(4), pp. 842-845, 1994.
- [6] Fossen, T. I., *Guidance and Control of Ocean Vehicles*, Wiley and Sons, New York, NY, 1994.
- [7] Funahashi, K., "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, 2, 183-192, 1989.
- [8] Girosi, F., and Poggio, T., "Networks and the best approximation property", *Biological Cybernetics*, 63, 169-176, 1990.
- [9] Hornik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators", *Neural Networks*, 2, 359-366, 1989.
- [10] Hughes, P. C., *Spacecraft Attitude Dynamics*, Wiley and Sons, New York, NY, 1986.
- [11] Paielli, R., and Bach, R., "Attitude control with realization of linear error dynamics", *J. Guid. Control and Dyn.*, 16(1), 182-189, 1993.
- [12] Papadopoulos, E. G., *On the Dynamics and Control of Space Manipulators*, Ph.D. Thesis, MIT Dept. of Mech. Eng., Cambridge, MA, 1990.
- [13] Papadopoulos, E. G., "On the nature of control algorithms for free-floating space manipulators", *IEEE Trans. Rob. Aut.*, 7(6), 750-758, 1991.
- [14] Poggio, T., and Girosi, F., "Networks for approximation and learning", *Proc. IEEE*, 78(9), 1481-1497, 1990.
- [15] Reed, J. S., and Ioannou, P. A., "Instability analysis and robust adaptive control of robotic manipulators", *IEEE Trans. Rob. and Aut.*, 5, 381-386, June 1989.
- [16] Sanner, R. M., *Stable Adaptive Control and Recursive Identification of Nonlinear Systems Using Radial Gaussian Networks*, PhD Thesis, MIT Dept. of Aeronautics and Astronautics, May 1993.
- [17] Sanner, R. M., and Slotine, J.-J. E., "Gaussian networks for direct adaptive control", *IEEE Trans. Neural Networks*, 3(6), pp 837-863, 1992.
- [18] Sanner, R. M., and Slotine, J.-J. E., "Stable adaptive control of robot manipulators using "neural" networks", to appear *Neural Computation*, 1995.
- [19] Sanner, R. M., and Vance, E. E., "Adaptive control of free-floating space robots using "neural" networks", submitted, 1995 *American Control Conference*, Sept 1994.
- [20] Sanner, R. M., and Vance, E. E., *SSL Report*, University of Maryland, Dept. of Aerospace Engineering, 1994 (in preparation).
- [21] Slotine, J.-J. E., and Di Benedetto, "Hamiltonian adaptive control of spacecraft", *IEEE Trans. Automatic Control*, 35, pp. 848-852, 1990.
- [22] Slotine, J.-J. E. and Li, W., *Applied Nonlinear Control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [23] Slotine, J.-J. E., and Li, W., "On the adaptive control of robotic manipulators", *Intl. Journal of Rob. Res.*, 6(3), 1987.
- [24] Wen, J. T.-Y., and Kreutz-Delgado, K., "The attitude control problem", *IEEE Trans Autom. Control*, 36(10), 1148-1162, 1991.
- [25] Wie, B., Weiss, H., and Arapostathis, A., "Quaternion feedback regulator for spacecraft eigenaxis rotations", *J. Guidance, Control, and Dynamics*, 12(3), 375-380, 1989.
- [26] Wie, B., and Barba, P.M., "Quaternion feedback for spacecraft large angle maneuvers", *J. Guidance, Control, and Dynamics*, 8(3), 360-365, 1985.

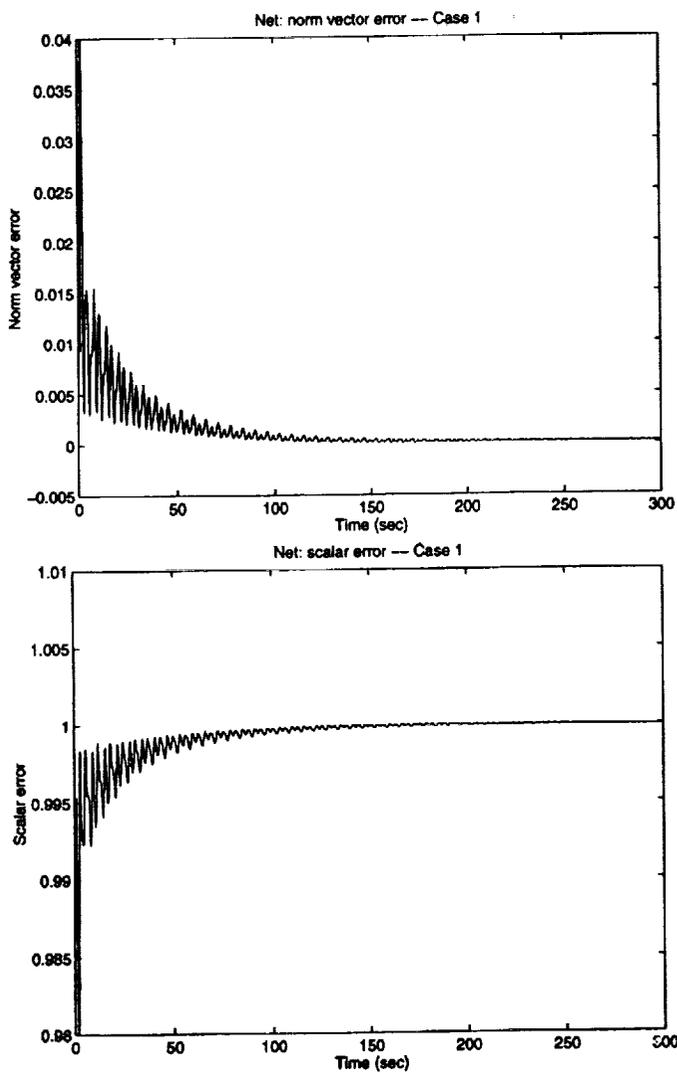


Figure 1: Attitude tracking performance using the proposed adaptive neurocontroller with the dynamics (1). The top figure shows the norm of the the vector part of the error quaternion,  $\|\tilde{\epsilon}\|^2$ , while the bottom figure shows the scalar part of the error quaternion,  $\tilde{\eta}$ .

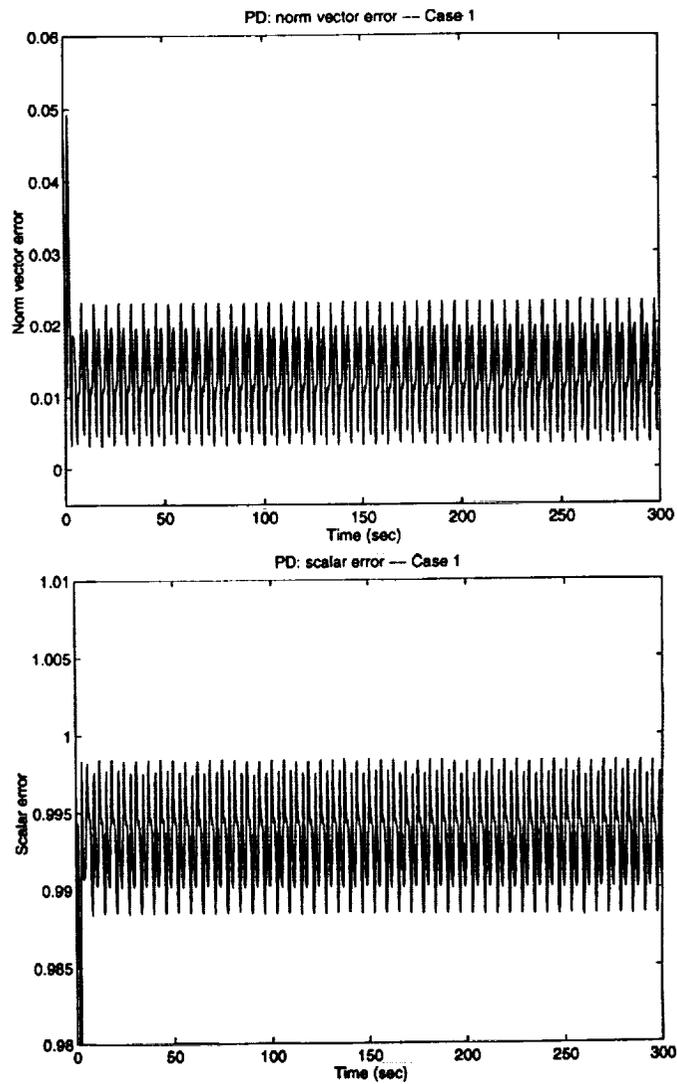


Figure 2: Attitude tracking performance with the dynamics (1) omitting the adaptive contribution of the neural network. The top figures shows the norm of the the vector part of the error quaternion,  $\|\tilde{\epsilon}\|^2$  while the bottom figure shows the scalar part of the error quaternion,  $\tilde{\eta}$ .

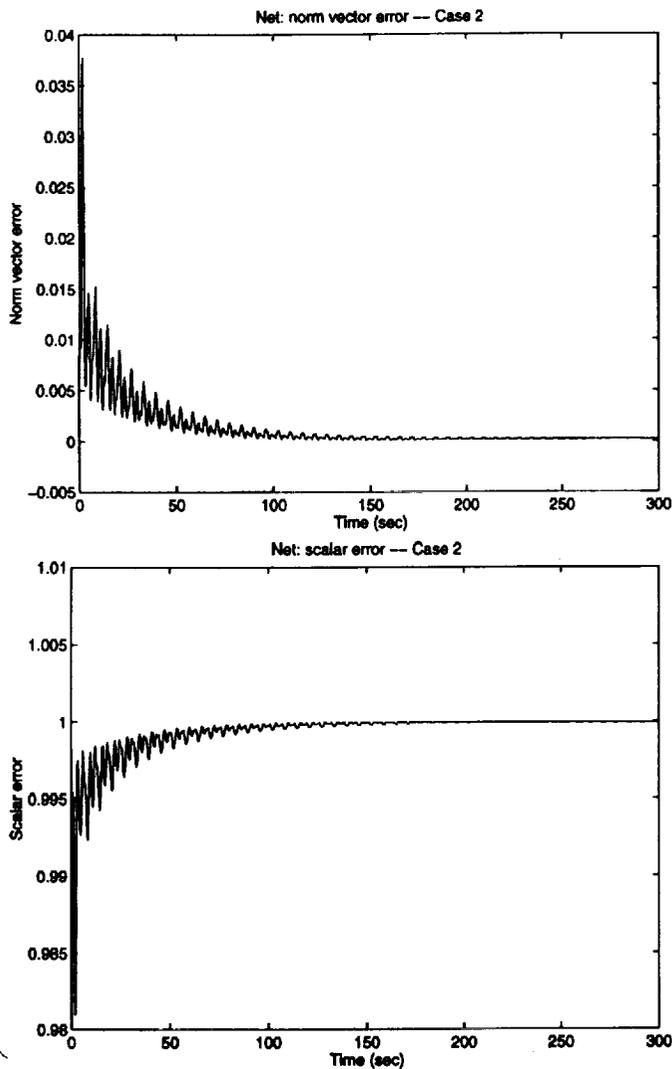


Figure 3: Attitude tracking performance using the proposed adaptive neurocontroller with the dynamics (14). The top figure shows the norm of the the vector part of the error quaternion,  $\|\tilde{\epsilon}\|^2$ , while the bottom figure shows the scalar part of the error quaternion,  $\hat{\eta}$ .

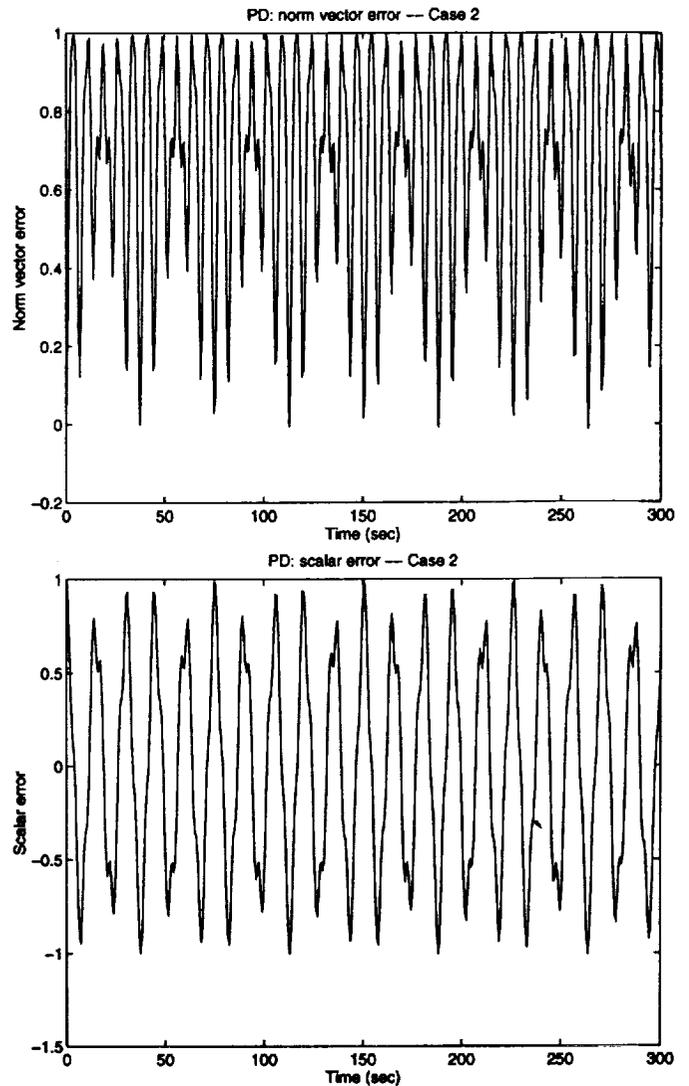


Figure 4: Attitude tracking performance with the dynamics (14) omitting the adaptive contribution of the neural network. The top figures shows the norm of the the vector part of the error quaternion,  $\|\tilde{\epsilon}\|^2$  while the bottom figure shows the scalar part of the error quaternion,  $\tilde{\eta}$ .

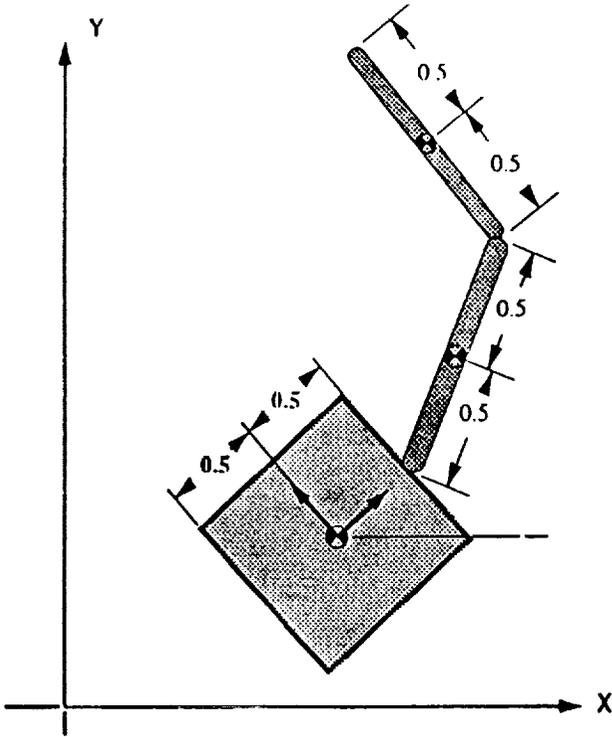


Figure 5: Diagram of the 3DOF simulation model

	spacecraft	link 1	link 2
mass (kg)	40	4	3
inertia (kg-m <sup>2</sup> )	6.667	.333	.250
length (m)	1	1	1

Figure 6: Physical parameters of the simulation model

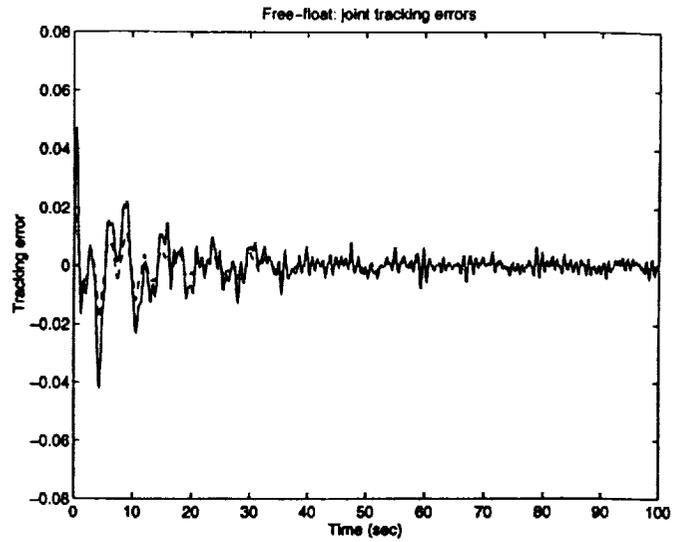


Figure 7: Joint angle tracking performance for the free-floating space robot using the proposed adaptive neurocontroller.

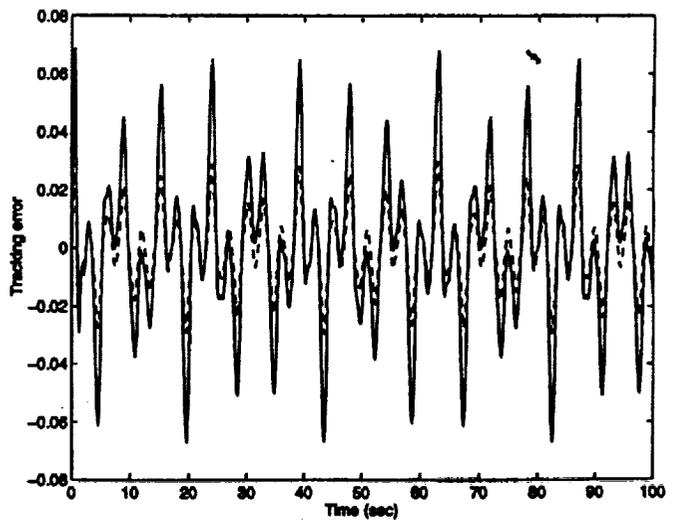


Figure 8: Joint angle tracking performance without use of the adaptive contribution of the neural network.